A **Deterministic Finite Automaton (DFA)** is a formal model of computation used in automata theory to represent and recognize patterns within input data, particularly strings. It is called *deterministic* because, for each state and input symbol, there is exactly one state to transition to.

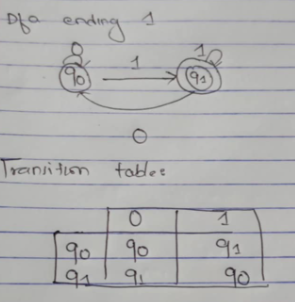
**Formal Definition of DFA:**

A DFA is typically represented as a 5-tuple:

⟨Q,Σ,δ,q0,F⟩

Where:

* Q is the set of states.
* Σ is the alphabet of input symbols.
* δ the transition function (δ:Q×Σ→Q)
* q0q\_0q0​ is the initial state.
* FFF is the set of accept states.



A **Pushdown Automaton (PDA)** is a theoretical model of computation that is similar to a finite automaton but with an added stack data structure. This stack allows the PDA to have more computational power than a finite automaton, making it capable of recognizing context-free languages, which cannot be recognized by finite automata.

**PDA Definition:**

A **Pushdown Automaton (PDA)** is defined as a 7-tuple:

PDA=(Q,Σ,Γ,δ,q0,Z0,F)

Where:

1. **Q** is a finite set of states.
2. **Σ** is a finite set of input symbols (the alphabet of the input).
3. **Γ** is a finite set of stack symbols (the alphabet of the stack).
4. **δ** is the transition function, where:

δ:Q×(Σ∪{ϵ})×Γ→P(Q×Γ∗)

1. **q₀** is the start state, where q0∈Qq₀ \in Qq0​∈Q.
2. **Z₀** is the initial stack symbol, where Z0∈ΓZ₀ \in ΓZ0​∈Γ.
3. **F** is the set of accepting states, where F⊆QF

